

## Using the Protection Object to Protect Your Worksheets

Michael Stowe  
Microsoft Corporation

April 2001

Applies to:  
Microsoft® Excel 2002

**Summary:** Microsoft Excel 2002 includes new objects that expand the existing capability to protect worksheets. This article discusses those new objects and demonstrates how to programmatically protect worksheets. (4 printed pages)

Download [ODC\\_xlwebProtSht.exe](#).

Download the ODC\_xlwebProtSht.exe sample file. (87 KB)  
ODC\_xlwebProtSht  
MSDNSamples\OfficeXP

### Contents

[Introduction](#)  
[The Protection Object](#)  
[Assigning User Permissions to Cell Ranges](#)

### Introduction

To protect your worksheet in Microsoft Excel prior to the 2002 version, you would lock the cells that you wanted to prevent users from changing. Unfortunately, your users could not perform basic tasks such as inserting rows or columns, or formatting the cells of that same worksheet.

Thankfully, the worksheet protection model in Excel 2002 is much richer than earlier versions. You can now allow users to perform basic worksheet tasks such as inserting and deleting cells while the worksheet is protected. You can assign permissions to specific cell ranges, allowing only certain users to edit specific portions of your worksheets.

This article illustrates how to use the methods and objects that make up the protection object model in Excel 2002 to perform the following tasks:

- Protect worksheets using the **Protect** method
- Assign certain users to specific ranges on a protected worksheet
- Auditing edit ranges that have been added to a worksheet

### The Protection Object

Each worksheet in a workbook contains a **Protection** object. The following table describes the properties of the **Protection** object.

**Table1. Protection object properties**

Property name	When True, allows user to:	Read-only?
<b>AllowDeletingColumns</b>	Delete columns on a protected worksheet.	Yes
<b>AllowDeletingRows</b>	Delete rows on a protected worksheet.	Yes
<b>AllowFormattingCells</b>	Format cells on a protected worksheet.	Yes
<b>AllowFormattingColumns</b>	Format columns on a protected worksheet.	Yes
<b>AllowFormattingRows</b>	Format rows on a protected worksheet.	Yes

<b>AllowInsertingColumns</b>	Insert columns on a protected worksheet.	Yes
<b>AllowInsertingRows</b>	Insert rows on a protected worksheet.	Yes
<b>AllowInsertingHyperlinks</b>	Insert hyperlinks on a protected worksheet.	Yes
<b>AllowFiltering</b>	Use AutoFilter on a protected worksheet.	Yes
<b>AllowSorting</b>	Sort cells on a protected worksheet.	Yes
<b>AllowUnsingPivotTables</b>	Use PivotTable reports on a protected worksheet.	Yes
<b>AllowEditRanges</b>	Not a <b>Boolean</b> property. Returns the <b>AllowEditRanges</b> collection.	N/A

With the exception of the **AllowEditRanges** property, the **Protection** object's properties are set when you use the **Protect** method to protect a worksheet.

### The Protect Method

The **Worksheet** object's **Protect** method has gained eleven new arguments. Each argument corresponds to the read-only properties in Table 1 above. Each new argument allows you to control whether or not the user can access some basic Excel features on a protected worksheet.

The following example protects each worksheet in the active workbook. The user is allowed to format and sort cells. By setting the **AllowFormattingCells** and **AllowSorting** arguments to **True**, the **AllowFormattingCells** and **AllowSorting** properties of each worksheet's **Protection** object is set to **True**.

Note that the default setting for each of the **Allow** arguments is **False**.

#### Sample 1. Protect all worksheets

```
Sub ProtectMethodAllSheets()

    Dim shtCurrent As Worksheet

    For Each shtCurrent In ActiveWorkbook.Worksheets
        shtCurrent.Protect Password:="Pass", Contents:=True, _
            DrawingObjects:=True, Scenarios:=True, _
            AllowFormattingCells:=True, AllowSorting:=True
    Next shtCurrent

End Sub
```

### Assigning User Permissions to Cell Ranges

Another new protection feature in Excel 2002 is the ability to assign user-level permissions to different regions on a protected worksheet. Within each edit range, you can specify the users who can edit the range without unlocking the entire worksheet. Additionally, you can specify that a user provide a range-specific password in order to make changes to the worksheet.

Each worksheet contains an **AllowEditRanges** collection that contains the collection of edit ranges for that worksheet. Use the **AllowEditRanges** property of the **Protection** object to return the collection of edit ranges. Use the **Add** method to add an **AllowEditRange** object to the worksheet.

The list of users for each **AllowEditRange** object is stored in the **UserAccessList** collection. The **Users** property of the **AllowEditRange** object is used to return the collection of users for the range. Use the **Add** method to add users to an edit range.

The AddEditRange procedure in the sample download illustrates how to create and add users to an edit range.

### Auditing Edit Ranges

You may want to add several edit ranges to each worksheet in an important workbook, each with a long list of users. Unfortunately, Excel does not provide an easy way to report all of the edit ranges in a workbook. The ListEditRangesAndUsers procedure in the sample download illustrates how to create a snapshot of the edit ranges in

the active workbook.

The `ListEditRangesAndUsers` procedure creates a report that contains the following items for each user that has been added to an edit range in the active workbook:

- The title of the edit range
- The cell address of the edit range
- The user's name
- A **Boolean** value indicating whether or not the user is allowed to make changes in the edit range without specifying a password

**Note** The password for the edit range is not included in the report. The reason is that you cannot derive the password of an edit range programmatically. You can use the **ChangePassword** method to change the password of an edit range if you know the password.

The example loops through each worksheet in the active workbook, and then loops through the **AllowEditRanges** collection for the worksheet. The code then loops through each user that has been added to the edit range and writes the information for each user to a new workbook.

## Conclusion

With the 2002 version, Excel now offers the protection features you need for greater security in the solutions you deploy to your users.

[Contact Us](#) | [E-Mail this Page](#) | [MSDN Flash Newsletter](#)

© 2002 Microsoft Corporation. All rights reserved. [Terms of Use](#) [Privacy Statement](#) [Accessibility](#)